

RTI Performance Framework

9 April 1997

Background

- **There is tremendous interest in ‘RTI Performance’**
 - **However, there has also been significant variation in what is meant by ‘RTI Performance’**
- **A common way to characterize RTI performance is needed to provide a common foundation for discussion across the HLA community (users, developers, planners)**

RTI Performance Framework

- **A performance framework is needed to...**
 - **Provide common terms of reference for discussions**
 - **Provide a “nutrition label” for RTIs**
- **This provides..**
 - **A common understanding of performance test results**
 - **Basis for comparison of RTI performance**
 - **between versions (e.g., between RTI F.0 and RTI 1.0)**
 - **between implementations (e.g., between RTI 2.0 and RTI 1.0)**
 - **between different federation applications**
 - **A context for procuring new RTI software**

Performance Framework Actions To Date

- **AMG-17 Action Item**
 - requested input from AMG members on RTI performance needs
- **Variety of inputs were received in response to the action item**
- **A draft RTI performance worksheet was created based on these inputs**
 - first step towards developing a framework
 - place performance needs into terms of RTI services and federation characteristics
- **Draft worksheet was sent out for comments and was used as basis for discussion at technical exchange**
- **Held a technical exchange meeting to review worksheet and discuss next steps**

Technical Exchange

- **Held on 3/18/97 at DMSO, with 16 participants**
 - All participants were realtime HLA users
 - Results of meeting focused on this user subset; input is needed from rest of HLA community
- **Technical exchange addressed three questions:**
 - Which RTI services have performance constraints for different applications?
 - What characteristics of HLA federations impact the performance of RTI services provision?
 - What specific resources have different users found to be useful in the past, which could be applied to the problem?
- **Included a presentation of current performance test approach taken by RTI IPT**

Technical Exchange Discussion Points

- **Focus of performance interest was on 'Update-reflect attribute' transactions**
 - Other areas were of known interest but focus was in this area
- **Recognition that there are a wide range of federation characteristics which can impact the performance of an RTI in this area**
 - Important that these are measured/controlled if results are to be generally useful
 - Revised worksheet reflects those identified
- **Different user communities have both different performance criteria and different federation characteristics**
 - Useful performance framework needs to provide right 'categories' but allow 'values' to be adjusted for needs of application
- **Any 'nutrition label' will only provide a 'bench mark' for potential RTI users; system specific testing will be required to ensure RTI meets specific application needs**

Results of Technical Exchange

- **Revised worksheet to reflect results of the discussions**
 - **Performance worksheet consists of four tables:**
 - **Federate/Federation Parameters**
 - **Performance Test Factors**
 - **Test Environment Conditions**
 - **RTI Services**
- **Re-cast RTI IPT performance testing in this framework**
 - **Jeff Olszewski to present RTI IPT performance experiments and results in terms of draft worksheet**

Federate/Federation Parameters

Parameters	Value(s)
Simultaneous federation executions	
Federates per federation	
Federates per host	
Objects per federate	
Number of attributes per object	
Average attribute updates per unit time	
Attribute size	
Number of parameters per interaction	
Average interactions per unit time	
Parameter size	
Publish/Subscribe topology	

Performance Test Factors

Factors	Value(s)
Application	
API used	
RTI version	
Delivery type	
Federate hardware	
Federate OS	
Network	
CPU utilization	
Test duration	
Time mgmt. scheme	

Performance Test Environment

Conditions	Value(s)
Logging software	
Software configuration	
RTI fedex host	
RTI exec hosts	
Workstation time synchronization	
Non-test network traffic	

RTI Services

SERVICE	IF Ref	Srvcs Used		SERVICE	IF Ref	Srvcs Used
Create Federation Execution	2.1	X		Delete Object	4.8	X
Destroy Federation Execution	2.2	X		Remove Object†	4.9	X
Join Federation Execution	2.3	X		Change Attribute Transportation Type	4.10	
Resign Federation Execution	2.4	X		Change Attribute Order Type	4.11	
Request Pause	2.5			Change Interaction Transportation Type	4.12	
Initiate Pause†	2.6			Change Interaction Order Type	4.13	
Paused Achieved	2.7			Request Attribute Value Update	4.14	
Request Resume	2.8			Provide Attribute Value Update†	4.15	
Initiate Resume†	2.9			Retract	4.16	
Resume Achieved	2.10			Reflect Retract†	4.17	

Federate/Federation Parameters - Jager Example

Federate/Federation Parameters

Parameters	Value(s)
Simultaneous federation executions	1
Federates per federation	2,4,6,8
Federates per host	1
Objects per federate	5,10
Number of attributes per object	7
Average attribute updates per unit time	as many as possible
Attribute size	4 bytes/attribute
Number of parameters per interaction	3 (max)
Average interactions per unit time	as many as possible
Parameter size	4 bytes/parameter
Publish/Subscribe topology	all-to-all

Performance Test Factors, Environment- Jager Example

Performance Test Factors

Factors	Value(s)
Application	Jager
API used	C++ 1.0
RTI version	F.0
Delivery type	best effort, reliable
Federate hardware	Sun Ultra 2's
Federate OS	Solaris 2.5
Network	ATM LAN
CPU utilization	below max
Test duration	~200 sec
Time mgmt. scheme	not time constrained, not time regulated

Test Environment

Conditions	Value(s)
Logging software	HLA testbed MOP Mgr.
Software configuration	standard MOP cfg.
RTI fedex host	federate A
RTI exec hosts	federate A
Workstation time synchronization	XNTP, polling at 1 minute intervals
Non-test network traffic	minimum, but not clean LAN